

In the Claims:

Please amend Claims 1, 2, 5, 7, 9, 11, 12, 15, 17 and 19; cancel Claims 21-27, 29 and 35-36; and add new Claims 37-42, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

1. (Currently Amended) A system for loading software applications, comprising:
a server for ~~providing~~ storing and running a plurality of software application applications, wherein each of said plurality of software application applications includes a plurality of deployable modules and classes associated therewith, and wherein the software applications can be customized by a software developer and then deployed to run on the same server;
a control file, that can be edited by ~~[[a]]~~ the software developer and associated with said plurality of software application applications, wherein said control file specifies a hierarchy of application classloaders to be used with the modules in said plurality of software application applications, and wherein the hierarchy includes a plurality of nested branches that are specified by the software developer to provide namespace separation between two or more of the plurality of software applications or between different modules in any one of the software application applications; and
a deployment ~~mechanism~~ utility that upon receiving a request to deploy and run a software application on the server, parses the control file and determines which classloaders are specified therein for the software application being deployed, and then loads with said software application onto the server a selection of said application classloaders corresponding to the hierarchy specified by said control file, including, if a particular software application or a module in said a software application is being redeployed then loading only the application classloaders that are specified in the branches for that particular software application or module, independently of without loading any of the other branches in the hierarchy.
2. (Currently Amended) The system of claim 1 wherein said control file can be modified by a software developer to specify a particular hierarchy of application classloaders to be used with a particular software application.

3. (Original) The system of claim 2 wherein said control file is a deployment descriptor.
4. (Original) The system of claim 3 wherein said control file is interpreted by an application container constructor during deployment so as to define the application container.
5. (Currently Amended) The system of claim 4 wherein said interpretation includes traversing the hierarchy and building parent child relationships between the tiers of selected application classloaders.
6. (Original) The system of claim 1 wherein said hierarchy is specified by a classloader structure declaration.
7. (Currently Amended) The system of claim 1 wherein a combination of said modules may be associated with a plurality of subordinate application classloaders.
8. (Canceled).
9. (Currently Amended) The system of claim 1 wherein the server provides multiple software applications, each with their own hierarchy of application classloaders.
10. (Canceled).
11. (Currently Amended) A method for loading software applications on a server, comprising the steps of:
providing a server for storing and running a plurality of software applications;
providing a plurality of software application applications, wherein each of said plurality of software application applications includes a plurality of deployable modules and classes associated therewith, and wherein the software applications can be customized by a software developer and then deployed to run on the same server;

parsing providing a control file associated with said software application, wherein said control file can be edited by a software developer and specifies a hierarchy of application classloaders to be used with the modules in said software application, and wherein the hierarchy includes a plurality of nested branches that are specified by the software developer to provide namespace separation between two or more of the plurality of software applications or between different modules in any one of the software application applications; and
upon receiving a request to deploy and run a software application on the server.

parsing the control file and determining which classloaders are specified therein for the software application being deployed.

retrieving a selection of said application classloaders according to the hierarchy specified by said control file[(:)], and

loading said modules and classes onto the server as part of said software application corresponding to said hierarchy, including, if a particular software application or a module in said a software application is being redeployed then loading only the application classloaders that are specified in the branches for that particular software application or module, independently of without loading any of the other branches in the hierarchy.

12. (Currently Amended) The method of claim 11 wherein said control file can be modified by a software developer to specify a particular hierarchy of application classloaders to be used with a particular software application.

13. (Original) The method of claim 12 wherein said control file is a deployment descriptor.

14. (Original) The method of claim 13 wherein said control file is interpreted by an application container constructor during deployment so as to define the application container.

15. (Currently Amended) The method of claim 14 wherein said interpretation includes traversing the hierarchy and building parent child relationships between the tiers of selected application classloaders.

16. (Original) The method of claim 11 wherein said hierarchy is specified by a classloader structure declaration.

17. (Currently Amended) The method of claim 11 wherein a combination of said modules may be associated with a plurality of subordinate application classloaders.

18. (Canceled).

19. (Currently Amended) The method of claim 11 wherein the server provides multiple software applications, each with their own hierarchy of application classloaders.

20-30. (Canceled).

31. (Previously Presented) The system of claim 1 wherein each of the plurality of modules is one of an EJB or Web application file, together with associated classes, configuration rules and resource files for that EJB or Web application file.

32. (Previously Presented) The system of claim 1 wherein the hierarchy that includes a plurality of branches specified by the software developer further comprises a plurality of nested references to modules and/or individual class files as specified by the software developer.

33. (Previously Presented) The method of claim 11 wherein each of the plurality of modules is one of an EJB or Web application file, together with associated classes, configuration rules and resource files for that EJB or Web application file.

34. (Previously Presented) The method of claim 11 wherein the hierarchy that includes a plurality of branches specified by the software developer further comprises a plurality of nested references to modules and/or individual class files as specified by the software developer.

35-36. (Canceled).

37. (New) A system for loading software applications, comprising:

an application server that includes a memory, for storing and running a plurality of Java software applications, wherein each of the plurality of Java software applications includes a plurality of deployable modules associated therewith, and wherein the Java software applications and modules can be customized by a software developer and then deployed to run on the server;

a deployment descriptor file that is used in deploying the Java software applications to the server,

wherein the deployment descriptor file is edited by the software developer to include a classloader structure definition, and

wherein the classloader structure definition specifies a nested hierarchy of application classloaders to be used in loading one or more Java software applications and modules, and

wherein the nested hierarchy of application classloaders provides a namespace separation between at least two or more of the plurality of Java software applications, or between different groups of modules in one of the software applications; and

a deployment utility that, upon receiving a request from the software developer to deploy a particular Java software application to the server,

parses the deployment descriptor file and determines which classloaders are specified therein for the particular Java software application being deployed, and

loads into the memory of the server only those classloaders that are identified in the namespace and nested hierarchy of the particular Java software application being deployed, including loading any application modules in the order specified in the nested hierarchy for that particular Java software application, and without requiring the loading of other modules or other applications branches in another namespace.

38. (New) The system of claim 37 wherein each of the plurality of modules is one of an EJB or Web application file, together with associated classes, configuration rules and resource files for that EJB or Web application file.

39. (New) The system of claim 37 wherein the hierarchy includes a plurality of branches within the hierarchy that are specified by the software developer to provide the namespace separation.

40. (New) The system of claim 37 wherein the deployment descriptor file is interpreted by the application server during deployment and is used to define the application container on the server.

41. (New) The system of claim 37 wherein the classloader structure definition specifies a nested hierarchy of application classloaders to be used in loading a first and a second Java software applications into memory, and wherein the nested hierarchy of application classloaders provides a namespace separation between the Java software applications, so that upon receiving a request from the software developer to deploy only the first of the Java software applications to the server, the system loads into the memory of the server only those classloaders that are identified in the namespace and nested hierarchy of the first Java software application being deployed, independently or without loading classloaders for the second Java software application.

42. (New) The system of claim 37 wherein the classloader structure definition specifies a nested hierarchy of application classloaders to be used in loading a first and a second group of modules within a single Java software application into memory, and wherein the nested hierarchy of application classloaders provides a namespace separation between the first and a second group of modules in the single Java software application, so that upon receiving a request from the software developer to deploy only the first of the group of modules to the server, the system loads into the memory of the server only those classloaders that are identified in the namespace and nested hierarchy of the first group of modules, independently or without loading classloaders for the second group of modules.